WHAT IS CLAIMED IS:

1. (Currently Amended) A USB host system operationally coupled to a computing system with a main processor, comprising:

a first processor that implements a USB driver and a host controller driver without using the main processor resources;

a downstream USB port; and,

a communication area directly accessible by both the main processor and the first processor such that the first processor interfaces with the main processor via the communication area using predefined records in pre-defined formats,

wherein the main processor writes a data transfer request in the communication area in a pre-defined record format, and

wherein the first processor schedules and completes the request via a USB host controller.

2. (Previously Presented) The USB host system of claim 1, wherein the communication area is a dual port memory.

3. (Previously Presented) The USB host system in claim 1, wherein the communication area consists of multiple FIFO registers.

4. (Previously Presented) The USB host system in claim 1, wherein an interrupt polled from a USB interrupt pipe is converted to an interrupt signal to the main processor.

5. (Previously Presented) The USB host system of claim 1, wherein the main processor interfaces with the host system via a standard microprocessor bus.

6. (Previously Presented) The USB host system in claim 1, wherein a hub is used to provide multiple downstream USB ports.

7. (Previously Presented) The USB host system of claim 1, wherein data in the communication area is directly sent out on a USB bus.

8. (Previously Presented) The USB host system in claim 1, wherein data received from the USB bus are written directly in the communication area.

9. (Previously Presented) The USB host system of claim 1, wherein the USB host system provides a USB function to the main processor.

10. (Previously Presented) The USB host system of claim 1, wherein the second processor runs an operating system supporting USB, and the USB host system provides a USB host function to the second processor by intercepting calls to a USB driver in the operating system.

11. (Currently Amended) A USB host system operationally coupled to a computing system, comprising:

a first processor implementing a function for managing a USB host controller with or without an operating system running on the computing system and implementing a USB driver and a host controller driver;

a downstream USB port; and,

an interface between the first processor and a second processor that provides a high-level USB pipe view of a USB system to an application program running on the second processor in the computing system,

wherein the interface comprises a memory that is directly accessed by both the first and second processors, and

wherein the second processor interfaces with the host system via a standard microprocessor bus.

12. (Cancelled).

13. (Cancelled).

14. (Previously Presented) The USB host system of claim 11, wherein a hub is used to provide multiple downstream USB ports.

15. (Previously Presented) The USB host system of claim 11, wherein the host system is used to provide a USB host function to the second processor.

16. (Previously Presented) The USB host system of claim 11, wherein the second processor runs an operating system supporting a USBD,

wherein the host system provides a USB host function to the second processor, including a USBD function, and

wherein the host system processes a USB transfer request by the second processor by intercepting calls to the USBD in the operating system and passing the calls to the USBD in the host system.

17. (Currently Amended) An information processing system comprising:

a first processor that implements a USB driver and a host controller driver;

a data transfer host system comprising a second processor implementing a first data transfer driver managing a data transfer between the first processor and a device;

a data transfer port for connecting the device to the data transfer host system; and,

an interface between the host system and the first processor that provides a high-level view of the data transfer process to the first processor,

wherein the interface comprises an area in a memory that is directly accessible by both the first processor and the second processor.

18. (Cancelled).

19. (Previously Presented) The information processing system of claim 17, wherein the second processor is used to reduce the number of interrupts to the first processor.

20. (Previously Presented) The information processing system of claim 17, wherein the second processor is used to reduce the frequency of interrupts to the first processor.

21. (Previously Presented) The information processing system of claim 17, wherein the first processor interfaces the data transfer host system via a standard microprocessor bus.

22. (Previously Presented) The information processing system of claim 17, wherein a hub is used to provide multiple ports for connecting a plurality of devices.

23. (Previously Presented) The information processing system of claim 17, wherein the first processor contains a second data transfer driver capable of managing the data transfer, and wherein a data transfer request by the first processor to the second data transfer driver is carried out by the data transfer host system.

24. (Currently Amended) A USB host comprising:

a first processor implementing a function of a USB system including a USB driver and a host controller driver;

a downstream USB port; and,

a memory connected to both the first processor and a second processor external to the USB host via a standard microprocessor bus interface,

wherein a first area of the memory with a first predetermined format is used for a first type of transfer, and a second area of the memory with a second predetermined format is used for a second type of transfer.

25. (Previously Presented) The USB host of claim 24, wherein a hub is connected to the downstream USB port so that multiple devices can be connected to the system.

26. (Cancelled).

27. (Previously Presented) The USB host of claim 24, wherein a third area of the memory with a third predetermined format is used for reporting device connection, enumeration and removal to the second processor.

28. (Previously Presented) The USB host of claim 27, wherein the third area is in a part of the memory that is read-only to the second processor.

29. (Previously Presented) The USB host of claim 24, wherein a fourth area of the memory with a fourth predetermined format is used for sending a USB command to the USB host.

30. (Previously Presented) The USB host of claim 24, wherein the starting address of each memory area for a transfer is used to identify the transfer.

31. (Previously Presented) The USB host of claim 24, wherein the second processor allocates the size of a memory area for a transfer to fit the need of the transfer.

32. (Previously Presented) The USB host of claim 24, wherein the second processor allocates the number of said areas to fit the need of a transfer.

33. (Previously Presented) The USB host of claim 24, wherein the respective starting addresses of the first and second areas are at different locations in the memory.

34. (Cancelled).

35. (Previously Presented) The USB host of claim 24, wherein the respective starting addresses of the first and second areas are at the same location in the memory.

36. (Previously Presented) The USB host of claim 24, wherein the predetermined formats of the first and second areas are the same.

37. (Previously Presented) The USB host of claim 24, wherein the respective starting addresses of the first and second areas are stored at fixed locations in the memory.

38. (Previously Presented) The USB host of claim 24, wherein the second processor writes a transfer request in one of said areas in the memory and notifies the first processor of the request with an interrupt signal.

39. (Previously Presented) The USB host of claim 24, wherein the first processor writes the status or data of a transfer into one of said areas in the memory and notifies the second processor of the request with an interrupt signal.

40. (Previously Presented) The USB host of claim 24, wherein a single format of the second area implements isochronous, interrupt and bulk transfers.

41. (Currently Amended) A USB host comprising:

a first processor implementing a function of a USB system <u>including a USB driver and a host controller driver</u>;

a downstream USB port; and,

a memory directly accessible by both the first processor and a second processor external to the USB host via a standard microprocessor bus interface,

wherein the second processor initiates a USB transfer by writing a transfer request and any data to be transferred into a first area in the memory, and

wherein the first processor carries out the transfer and writes the status of the transfer and any transferred data into a second area in the memory.

42. (Previously Presented) The USB host of claim 41, wherein a hub is connected to the downstream USB port so that multiple devices can be connected to the system.

43. (Cancelled).

44. (Previously Presented) The USB host of claim 41, wherein the first and second areas in the memory are the same area.

45. (Previously Presented) The USB host of claim 41, wherein the first and second areas in the memory use the same predefined format.

46. (Previously Presented) The USB host of claim 41, wherein the second processor runs an operating system that supports a USB driver, and wherein a USB transfer request initiated by the second processor to the USB driver is carried out by the USB host.

47. (Previously Presented) The USB host of claim 41, wherein the USB host transmits an interrupt signal to the second processor to notify the second processor that the transfer has been completed.

48. (Previously Presented) The USB host of claim 41, wherein the second processor transmits an interrupt signal to the USB host to notify the USB host that the second processor has initiated a USB transfer.

49. (Currently Amended) A USB host system operationally coupled to a computing system with a main processor, comprising:

a processor that interfaces with the main processor via a communication area using pre-defined records in pre-defined formats and implements a USB driver and a host controller driver,

wherein the main processor writes a data transfer request in the communication area in a pre-defined record format and the processor schedules and completes the request via a USB host controller, and

wherein the main processor and the processor are operationally coupled via a standard microprocessor bus interface.

50. (Previously Presented) The USB host system of claim 49, wherein the processor returns status and data to the main processor based on a request from the main processor.

51. (Previously Presented) The USB host system of claim 49, wherein the communication area is a dual port memory with plural registers.

52. (Previously Presented) The USB host system of claim 49, wherein the main processor may poll the communication area and/or be notified by an interrupt generated by the processor.

53. (Previously Presented) The USB system of claim 49, wherein the communication area is divided into a first area with a predefined format for a first type of transfer, and a second area with a second predefined format for a second type of transfer.

54. (Previously Presented) (Cancelled).

55. (Previously Presented) The USB host system of claim 49, wherein the communication area includes a third area with a third pre-defined format to report device connection and/or removal from the main processor.

56. (Previously Presented) The USB host system of claim 49, wherein the communication area includes a fourth area for storing a USB command in a fourth pre-defined format for sending a USB command to the processor.

57. (Previously Presented) The USB host system of claim 49, wherein the starting address of the communication areas are used to identify a transfer.

58. (Previously Presented) The USB host system of claim 49, wherein the main processor may allocate the dual port memory areas for a transfer.

59. (Previously Presented) The USB host system of claim 51, wherein the dual port memory may implement isochronous, interrupt and/or bulk transfers.